MCP SERVER | NO CODE | CLOUD HOSTED

Linear MCP

Track issues, projects, and sprints via conversation.

Linear (Issue Tracking & PM) lets your AI agent take full control of your product development lifecycle. You can track issues, audit team projects, and monitor sprint cycles—all through natural conversation. It's designed for teams that need to manage complex workflows without leaving their chat client.

A+ Quality Score 100/100

issue-tracking | sprint-planning | roadmap-management | task-management | workflow-automation | team-collaboration

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/linear-issue-tracking-pm

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Linear (Issue Tracking & PM) MCP

14 tools available
Cloud-hosted on Vinkius

Managing a product roadmap means juggling status updates, project scopes, and assignment changes across multiple boards. This MCP connects your agent directly into Linear, giving you full visibility into your development workflow. Instead of opening the browser and clicking through tabs just to find an issue's priority or its assigned team, you ask your AI client what you need. Your agent handles the complexity—it knows which labels mean 'urgent bug,' it can list every active project, and it tracks who owns what work across different teams. Connecting this MCP via Vinkius means you get access to Linear's entire data set from any compatible AI client. You use natural language to audit sprint cycles, pinpoint specific issues for deep context inspection, or even update a status directly. It's about having the data available instantly when your team needs it most.

Core Capabilities

01 — Audit Project and Team Structure

List all active projects, map out logical teams, and view global metadata tags to understand how work is categorized.

02 — Inspect Issue Details and Context

Search for specific issues using keywords or retrieve full descriptions, assigned labels, and internal priority levels instantly.

03 — Manage Task Status and Comments

Update an existing issue's status or add comments to notify assignees without opening the Linear application.

04 — Monitor Sprint Timelines

Get clear boundaries for active sprint cycles, including start/end dates and current progress counts.

05 — Create New Work Items

Generate a new issue within a specific team using the agent's context.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/linear-issue-tracking-pm** — connect your AI agent in three steps.

01 Subscribe to this MCP and enter your Linear API Key.

02 Connect it to your preferred AI client (Claude, Cursor, etc.).

03 Ask your agent a natural language question about issues, projects, or sprints.

The bottom line is you talk to your agent, and the data from Linear shows up in your chat window.

Built For

This MCP serves anyone who lives in a product board. It's for the Product Manager drowning in weekly sync prep, the Software Engineer needing quick context checks inside their IDE, or the Engineering Lead responsible for resource allocation across multiple projects.

Product Manager

Monitoring project progress and auditing sprint cycles to prepare materials for roadmap planning meetings.

Software Engineer

Fetching detailed issue contexts or updating task statuses directly from their coding environment without leaving the IDE.

Engineering Lead

Tracking team assignments and auditing issue distribution across multiple projects to ensure resources are allocated optimally.

What Changes When You Connect

01 Stop manually checking status boards. You can use the agent to search for issues or get deep context on any tracking ID instantly. This is way faster than copy-pasting IDs into a spreadsheet.

02 Audit your team's capacity easily. The MCP lets you list all active projects and monitor current sprint cycles, giving project leads immediate scope visibility without leaving their chat window.

03 Update work items with precision. Instead of navigating to an issue just to change its status, the agent handles transitions using `list_workflow_states` and then updates the item for you.

04 Know who owns what work. You can list all workspace members or check team boundaries (`list_teams`) so you know exactly which role is responsible for a given task.

05 Keep stakeholders informed without friction. Use `create_comment` to add status notes directly to an issue, ensuring everyone assigned gets real-time notifications.

Real-World Applications

Weekly Sprint Review Prep

A Product Manager needs a report on the current sprint. They ask their agent for the end date of the cycle and how many issues are 'To Do' or 'In Progress.' The agent uses `list_cycles` and aggregates the data, giving them an instant summary for their meeting agenda.

Triage a High-Priority Bug

A Software Engineer gets a bug report (BUG-404). Instead of opening Linear, they ask their agent for the issue's deep context. The agent uses `get_issue` to pull priority, labels, and full descriptions right into the IDE chat.

Reassigning Blocked Work

An Engineering Lead notices a critical task is stalled in 'In Progress.' They ask their agent to check which users are assigned to the project. The agent uses `list_users` and then facilitates re-routing by updating the issue status.

Project Scope Check

A PM needs to know if a new feature fits within an existing project scope. They ask their agent to list all active projects (`list_projects`) and check the details of the target project using `get_project`.

Patterns to Avoid

Trying to change status without knowing states

✗ AVOID

The user tries to tell the agent, 'Change issue ENG-105 to finished.' The agent fails because it doesn't know if 'finished' is a valid state ID.

✓ INSTEAD

First, ask your agent to use `list_workflow_states`. Once you confirm the correct state name (e.g., 'Done'), then tell the agent to execute `update_issue`.

Asking for a specific issue without context

✗ AVOID

The user just says, 'Show me that bug.' The system can't find it because Linear has thousands of issues.

✓ INSTEAD

Use the `search_issues` tool and provide keywords or assignees. For example, ask to search for issues related to 'API Timeout' assigned to John.

Creating an issue in the wrong team

✗ AVOID

The user asks to create a new bug but forgets which team owns that specific feature line.

✓ INSTEAD

First, use `list_teams` to confirm the correct team ID. Then instruct the agent to run `create_issue`, specifying the correct team.

The Right Fit

Use this MCP if your job requires frequent context switching between reading issue details, checking project scopes, and updating task statuses across Linear. If you live by sprint cycles and constantly cross-reference who is doing what work, this is for you. Don't use it if all you need is a simple list of users; in that case, just using the `list_users` tool directly works fine. However, if your primary goal is deep reporting or complex data modeling (like creating custom charts), you might be better off using a dedicated analytics dashboard tool instead of an MCP agent.

The Manual Grind of Product Development

Today, updating project status feels like a bureaucratic scavenger hunt. You open Linear, navigate to the specific board, find the ticket ID, check its current state, copy that ID into Slack for a teammate, and then maybe switch tabs entirely just to confirm which team owns the scope. It’s constant clicking, copying, and cross-referencing.

With this MCP, you talk to your agent instead of clicks. You simply ask, 'What's the status of ENG-105?' The agent instantly pulls that deep context—including its priority, labels, and current assignee—and presents it right where you are working.

Control Your Workflow with Linear MCP

Forget logging into multiple views just to get a full picture. You don't have to manually list every active project, check the current sprint boundaries, and then verify which team owns that work scope. The agent handles all those lookup steps behind the scenes.

Now you can audit your entire product roadmap using plain English conversation. It changes everything by making Linear’s massive data set immediately actionable.

Linear (Issue Tracking & PM) with 14 Tools

These tools let your agent perform specific actions within Linear, giving you precise control over issues, projects, teams, and workflows.

#	TOOL	DESCRIPTION
01	`list_teams`	Retrieves a list of all logical teams that map workspace access.
02	`create_issue`	Generates and creates a new issue within a specific Linear team, requiring the team ID.
03	`search_issues`	Searches for issues across Linear using keywords and returns matching status and assignment details.
04	`list_issues`	Lists recent, general issues from the entire Linear workspace.
05	`get_issue`	Gets detailed information for a single, specific issue tracking ID.
06	`list_projects`	Lists all active projects available in the workspace for monitoring scope.
07	`list_cycles`	Displays current sprint cycle bounds, including their start and end dates.
08	`list_labels`	Provides a list of global metadata tags used to categorize issues within the workspace.
09	`list_users`	Lists all explicitly mapped members in the workspace, confirming active access limits.
10	`get_viewer`	Authenticates and retrieves data for the current global user boundaries.
11	`update_issue`	Changes the status or details of an existing Linear issue.
12	`list_workflow_states`	Lists all valid workflow states (like Backlog, To Do, Done) to guide status changes.
13	`get_project`	Retrieves detailed information about one specific Linear project.
14	`create_comment`	Adds a new comment thread to an issue, notifying assignees and subscribers.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U List all active issues assigned to me in the 'Engineering' team

I've found 4 active issues for you in Engineering: 1) 'Fix API Timeout' (ENG-101), 2) 'Implement Auth Hook' (ENG-105), 3) 'Update Schema Docs' (ENG-108), and 4) 'Review PR #45' (ENG-112). Which one would you like to update the status for?

U Show me the details for issue 'ENG-101'

Issue ENG-101 ('Fix API Timeout'): Priority: Urgent, Status: In Progress, Label: 'Bug'. Description: 'API calls to the /v1/search endpoint are timing out after 30s under heavy load'. It is currently assigned to you and part of the current cycle.

U What is the end date for the current sprint cycle?

The current sprint cycle for the Engineering team ends this Friday, March 27th. There are currently 12 active issues in the cycle, with 8 in the 'Todo' or 'In Progress' states. Would you like a list of remaining tasks?

Frequently Asked Questions

01 **How do I use Linear MCP to find an issue's full context?**

You use the `get_issue` tool and provide a specific tracking ID. The agent then returns the priority, labels, and complete description for that single item.

02 **Can I update an issue status using Linear MCP?**

Yes, you can. You tell your agent to run `update_issue`, and it manages the state transition after confirming valid states via `list_workflow_states`.

03 **What is the best way to check my team's available resources?**

Use the `list_users` tool. This shows all explicitly mapped workspace members, helping you see who has active access and can take on new tasks.

04 **How does Linear MCP help with sprint planning?**

It helps by providing current tracking sprint cycle bounds using `list_cycles`. You get the start/end dates and a count of issues in various states for accurate planning.

05 **Do I need to know the team ID to create an issue with Linear MCP?**

Yes. To run `create_issue`, you first use `list_teams` to get the correct logical team segment boundary mapping before telling the agent where to place the new task.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"linear-issue-tracking-pm": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Linear (Issue Tracking & PM) is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Linear (Issue Tracking & PM). All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Linear (Issue Tracking & PM) MCP
Server ID	`019d75c7-840c-7162-9762-7843b4b55c40`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/linear-issue-tracking-pm.